Automated Annotation of Fungal and Algal Genomes.

I. Introduction

JGI annotation of fungal and algal genomes involves: automated gene modeling, manual inspection for quality control, and community curation of the predicted genes. Each pipeline produces structural and functional annotations of protein-coding and non-coding genomic features. The annotations are uploaded into a relational database, from which they are retrieved via a JGI Genome Portal for quality assessment and community annotation. Web-based visualization and analysis tools connect to the databases and pipeline log files to allow real-time monitoring and control of both the annotation process and the data it generates. Data- and user-management tools facilitate the data release process, data distribution to individual users, and submission of the data to GenBank.

The portals are collected into MycoCosm (for fungi [Grigoriev et al., 2014], https://mycocosm.jgi.doe.gov/) or PhycoCosm (for algae [Grigoriev et al., 2021], https://phycocosm.jgi.doe.gov/), allowing comparative and multi-omic analyses.

II. Requirements

The JGI Annotation Pipeline takes as input a genomic assembly, transcriptomic reads and assemblies, public protein databases, and configuration parameters for project-specific customization.

III. Procedure

The key steps of the JGI Annotation Pipeline are gene prediction, functional annotation, and comparative analysis.

A. Gene Modeling

The complex organization and gene structure of eukaryotic genomes pose challenges to gene prediction. The Pipeline comprises several stages: repeat-masking, gene prediction assisted with transcriptome and homologs from phylogenetically related species using different prediction methods, and validation of predicted gene models with several lines of evidence.

1. Repeat-masking. Before gene prediction, genomic scaffolds are masked using RepeatMasker [Smit et al. 2010] and a fungal or algal-specific library of repeats built from: 1) the standard RepBase library [Jurka et al. 2005], 2) the most frequent (>150x) repeats recognized by RepeatScout [Price et al, 2005], and 3) manually curated sets of transposons when available.

2. Mapping RNAs and proteins. All transcriptomic data for a given organism, either sequenced in-house or retrieved from GenBank or collaborator collections, are trimmed, clustered, and assembled into consensus sequences or contigs using appropriate sequencing platform-specific

transcriptome procedures outside of the Annotation Pipeline. These RNA reads and contigs are mapped to the genome assembly using BLAT [Kent, 2000], filtered with thresholds of 95% nucleotide identity and 80% coverage over sequence length, and used in gene modeling, model selection, and validation steps. RNA reads are also mapped using GMAP [Wu et al. 2005].

Proteins of related species and found in public databases such as NR (http://www.ncbi.nlm.nih.gov/BLAST/) are grouped taxonomically and mapped onto the masked genome assembly using BLASTx [Altschul et al. 1990] with e-value < $1xe^{-5}$. These alignments serve as seeds for homology-based gene prediction.

3. Modeling genes. Using the masked assembly, the Pipeline next deploys several gene prediction programs of 3 general types:

- *ab initio* modelers trained for the given genome: FGENESH [Salamov and Solovyev 2000]; GeneMark [Ter-Hovhannisyan et al, 2008].
- homology-based modelers using protein seeds: FGENESH+ [Salamov and Solovyev 2000]; GeneWise [Birney et al, 2004].
- transcriptome-based modelers using RNA mappings: EST_map (http://www.softberry.com/); combest [Zhou, 2015]; estExt (I. Grigoriev, unpublished).

4. Training gene predictors. To train FGENESH, the Pipeline automatically generates full-length gene models from RNA contigs, and then screens these models for completeness, quality, and redundancy. The Pipeline then combines this set with the set of GeneWise and FGENESH+ gene models. This combined set is randomly split into training and test subsets in the proportion 4:1. The models from the training subset provide hexamer frequencies derived from coding sequence (CDS) while intron structure informs exon/intron transition probabilities of the FGENESH parameter file. These newly derived parameters are tested on the test subset in parallel to the parameters created earlier for other genomes, and the best performing parameter set assessed by specificity and sensitivity of exon predictions is used for each genome. If either specificity or sensitivity of the best parameter set drops below 50%, we perform manual training.

The Pipeline uses the self-training version of GeneMark, which captures intron structures specific to fungal genomes.

5. Improving gene models. Since all gene predictors model only CDSs and not untranslated regions (UTRs), the Pipeline program estExt employs RNA contigs overlapping with gene models to determine gene UTRs, and to correct gene structures that disagree with transcript mappings.

GeneWise models are extended by finding in-frame upstream start and downstream stop codons. GeneWise models that include frameshifts are treated as potential pseudogenes.

6. Predicting non-coding genes. In addition to protein-coding genes, the Pipeline predicts non-coding genes. tRNAs are predicted using tRNAscan-SE [Lowe and Eddy 1997]. RFAM

non-coding RNA families are predicted using the Infernal software suite (S. Eddy, Janelia Farms). Regions of nucleotide conservation between genomes may suggest additional non-coding genes if supported by expression data; the Pipeline calculates conservation by VISTA whole-genome alignments [Ratnere & Dubchak, 2009].

B. Functional Annotation of proteins

All predicted gene model proteins are functionally annotated using SignalP [Nielsen et al. 1997] for signal peptides, TMHMM [Melen et al. 2003] for transmembrane domains, and InterProScan [Quevillon et al, 2005] for integrated collection of functional and structural protein domains, including PFAM. All proteins are also queried using BLASTp [Altschul et al. 1990] against NR, SwissProt (http://www.expasy.org/sprot/), KEGG [Kanehisa et al. 2006], and KOG [Koonin et al. 2004].

C. Gene model filtering and the Gene Catalog

1. Filtering gene models. The multiple gene predictors produce multiple overlapping gene models for each locus. The Pipeline excludes from further analysis those models that are similar to transposable element (TE) proteins, that have TE PFAM domain families, or that lie within repeat-masked regions. To select the best representative gene model we employ a heuristic approach (A. Salamov, unpublished) based on a combination of protein homology and transcriptome support. Homology support is measured by alignments with the best BLASTp hit from NR and other protein databases, where only alignments with BLASTp score > 50 and that cover at least 25% of length of gene models are considered. Transcriptome support is measured by correlation coefficient (CC) of the gene model relative to experimentally known, validated gene models, in this case mapped transcripts overlapping with the gene model. CC values range from 1 for perfect match between transcripts and predicted gene model to -1 for complete disagreement. Thus a gene model's transcriptome support is defined as the average of all CCs computed for each overlapping transcript. Each gene model is assigned the following empirical score: $S = S_{blast} * (cov_1 * cov_2 + CC_a)$, where S_{blast} is the combined BLASTp score of alignments between the gene model and its protein homolog, cov_1 and cov_2 are alignment coverages for the model and homolog respectively ($0 <= cov_1, cov_2 <= 1$), and CC_a is the average CC between the model and overlapping transcripts. At each locus, a model with the highest score is selected, and all other models, including those which have at least 5% CDS overlap with the selected model, are discarded.

The selected gene models form the GeneCatalog, which is subject to further genome analysis, manual curation, and, ultimately, GenBank submission.

2. Building deflines. The Pipeline builds a defline for each Gene Catalog protein by transferring text either from the top BLASTp protein hit with amino acid sequence identity > 80% and

alignment coverage > 80%, or from InterPro domains with e-value < $1e^{-15}$. When a defline does not meet GenBank requirements, it is replaced with 'hypothetical protein'.

3. Controlling quality of annotation. The Gene Catalog is assessed by multiple lines of evidence, including transcriptome and homology support, statistical metrics such as gene length and number of exons, completeness based on CEGMA [Parra et al. 2007], comparison with previously annotated genomes, and manual inspection. Quality assessment of annotated genome involves a multi-tier process that include (i) assessment by annotator, (ii) peer review, (iii) community annotation, and (iv) GenBank review.

D. Comparative analysis between genomes

The Pipeline subjects each annotated genome to comparative analyses at the assembly nucleotide level and the gene-protein level.

1. Genome level. The masked assembly is aligned to masked assemblies of related organisms using VISTA [Ratnere & Dubchak, 2009]. These alignments are rendered by the Genome Portal into regions of DNA conservation displayed on the genome browsers, and into syntenic regions visualized as interactive dotplots.

2. Gene/protein level. The Pipeline performs functional annotation on individual genes in the Gene Catalog by applying different classification schemes:

- GO terms (Ashburner et al, 2000; http://www.geneontology.org/) are assigned by mapping from InterPro domains and SwissProt hits.
- EC numbers (http://www.expasy.org/enzyme/) and placements in metabolic pathway maps are inferred from KEGG hits.
- KOG categories [Koonin et al. 2004] are assigned from KOG hits.
- Secondary metabolism clusters and classifications are inferred from an in-house procedure based on PFAM domains and physical proximity on the genome (A. Salamov, unpublished).
- CAZyme classifications are assigned outside of JGI by the Carbohydrate-Active enZYmes Database (http://www.cazy.org/) in a special collaboration.
- Peptidase classifications are assigned from BLASTp query of MEROPS (http://merops.sanger.ac.uk/).
- Transporter classifications are assigned from BLASTp query of TCDB (https://www.tcdb.org/).
- Transcription factor (TF) classifications are inferred from an in-house procedure base on a manually curated set of fungal and algal TF PFAM domains (A. Salamov, unpublished).

The functional annotations are profiled as tables of gene counts in each functional category.

Multigene families are predicted with the Markov clustering algorithm (MCL, [Enright et al. 2002]), which clusters proteins based on BLASTp alignment scores between them. Gene families are annotated using PFAM domains detected in cluster member sequences. Portal display of member gene structure, domain composition, and synteny assists validation of gene families. Gene families can in turn be used to assess the quality of new Gene Catalogs of subsequently annotated genomes of related species. The gene families shared between phylogenetic groups of genomes are also used to build the interactive Tree tools of MycoCosm and PhycoCosm.

IV. Implementation

The JGI Annotation Pipeline relies on a framework of pipeline infrastructure tools to monitor and control abstract pipelines running on Linux clusters. These tools enable automatic setup of the pipeline, visualization of the pipeline run with interactive control by annotators, APIs to external tools for gene family/cluster analysis and VISTA alignments, and automatic Genome Portal construction and configuration. Detailed error detection and fault identification procedures are provided for the annotators to troubleshoot problems.

A. Pipeline/Portal setup process. A Perl script controls the annotation input data, creates directories, copies files to appropriate locations, and builds and launches the pipeline. It also sends work requests to clustering (gene family), VISTA, and Portal subsystems.

B. Annotation process uses the Pipeline infrastructure to run a network of programs that implement the database and perform analyses required to create a genome portal. Extensive checking is done to ensure that programs are run correctly and log files are created for possible multiple runs of the pipeline. Pipelines can run on all the JGI compute clusters and can easily be moved from cluster to cluster as the workload changes and a cluster becomes busy.

C. Cluster Analysis (gene family) subsystem is notified by the Annotation Pipeline when data is available for multiple compared genomes. At the end, the Annotation Pipeline verifies completeness of the work and connects the clustering results to the organism database.

D. VISTA Analysis subsystem is notified by the Annotation Pipeline when the masked genome is made available, which also launches whole-genome synteny analysis. At the end of the pipeline, VISTA results are connected to the organism database automatically.

E. Genome Portal construction and configuration is initiated by the Annotation Pipeline, which also adds data and services to the Portal as corresponding analysis steps are executed to allow immediate display of results. At the end of the Annotation Pipeline the Portal is notified that the genome database is ready to move from the staging server to the production server.

F. Pipeline infrastructure tools. The pipeline construction web interface enables graphical construction of abstract pipelines that are configured by 1) symbolic on/off tags to include/exclude pipeline components (e.g. specific gene predictors) and by 2) text substitution macros that control

command lines generated within the pipeline. Pipeline sections are defined in a template which defines a program execution subgraph. Templates are nested to provide modularity in pipeline specification. The pipeline monitoring/control web interface allows annotators to view the pipeline graph with color coded status of pipeline modules. A mouse click on a program node in the graph instantly displays the log file for that program allowing quick problem determination. Parts of the pipeline can be suspended or moved to different queues on the compute cluster. The status of hardware can be also displayed.

V. References:

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990 Oct 5;215(3):403-10. doi: 10.1016/S0022-2836(05)80360-2. PMID: 2231712.

Ashburner, M. et al. Gene ontology: tool for the unification of biology. Nature Genetics 25, 25-9 (2000).

Bendtsen, J. D., Nielsen, H., von Heijne, G. & Brunak, S. Improved prediction of signal peptides: SignalP 3.0. Journal of Molecular Biology 340, 783-795 (2004).

Birney, E., Clamp, M. & Durbin, R. GeneWise and Genomewise. Genome Research 14, 988-995 (2004).

Enright AJ, Van Dongen S & Ouzounis CA. An efficient algorithm for large-scale detection of protein families. Nucleic Acids Research 30(7):1575-1584 (2002).

Grigoriev IV, Hayes RD, Calhoun S, Kamel B, Wang A, Ahrendt S, Dusheyko S, Nikitin R, Mondo SJ, Salamov A, Shabalov I, Kuo A. PhycoCosm, a comparative algal genomics resource. Nucleic Acids Res. 49(:D1004-D1011 (2021).

Grigoriev IV, Nikitin R, Haridas S, Kuo A, Ohm R, Otillar R, Riley R, Salamov A, Zhao X, Korzeniewski F, Smirnova T, Nordberg H, Dubchak I, Shabalov I. MycoCosm portal: gearing up for 1000 fungal genomes. Nucleic Acids Res. 42:D699-704 (2014).

Jurka J, Kapitonov VV, Pavlicek A, Klonowski P, Kohany O & Walichiewicz J. Repbase Update, a database of eukaryotic repetitive elements. Cytogenet. Genome Res. 110:462–467 (2005).

Kanehisa M, G. S., Hattori M, Aoki-Kinoshita KF, Itoh M, Kawashima S, Katayama T, Araki M, Hirakawa M. From genomics to chemical genomics: new developments in KEGG. Genome Biology 5, R7 (2006).

Kent WJ. BLAT--the BLAST-like alignment tool. Genome Res. 12(4):656-664 (2002).

Koonin, E. V. et al. A comprehensive evolutionary classification of proteins encoded in complete eukaryotic genomes. Genome Biology 5, R7 (2004).

Kuo A, Bushnell B, Grigoriev IV. Fungal genomics: sequencing and annotation, p 1–52. In Martin F (ed), Fungi. Advances in botanical research. Elsevier Academic Press, Cambridge, United Kingdom. (2014).

Lowe TM, Eddy SR. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. Nucleic Acids Res. 25(5):955-964 (1997).

Melen K, Krogh A & von Heijne G. Reliability measures for membrane protein topology prediction algorithms. J. Mol. Biol 327(3):735-744 (2003).

Nielsen H, Engelbrecht J, Brunak S & von Heijne, G. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. Protein Engineering 10:1-6 (1997).

Parra G, Bradnam K, Korf I. CEGMA: a pipeline to accurately annotate core genes in eukaryotic genomes. Bioinformatics. 2007 May 1;23(9):1061-7. doi: 10.1093/bioinformatics/btm071. Epub 2007 Mar 1. PMID: 17332020.

Price AL, Jones NC, Pevzner PA. De novo identification of repeat families in large genomes. Bioinformatics 21 Suppl 1:i351-8(2005).

Quevillon, E. et al. InterProScan: protein domains identifier. Nucleic Acids Res 33, W116-20 (2005).

Ratnere I, Dubchak I. Obtaining comparative genomic data with the VISTA family of computational tools. Curr Protoc Bioinformatics. 2009 Jun;Chapter 10:Unit 10.6.

Salamov, A. A. & Solovyev, V. V. Ab initio gene finding in Drosophila genomic DNA. Genome Res. 10, 516-22 (2000).

Smit, AFA, Hubley, R & Green, P. *RepeatMasker Open-3.0*. 1996-2010

Ter-Hovhannisyan V, Lomsadze A, Chernoff YO, Borodovsky M. Gene prediction in novel fungal genomes using an ab initio algorithm with unsupervised training. Genome Res. 18(12):1979-90 (2008).

Wu TD, Watanabe CK. (2005). GMAP: a genomic mapping and alignment program for mRNA and EST sequences. Bioinformatics, 21, 1859-75.

Zhou K, Salamov A, Kuo A, Aerts AL, Kong X, Grigoriev IV. Alternative splicing acting as a bridge in evolution. Stem Cell Investig. 2015 Oct 30;2:19. doi: 10.3978/j.issn.2306-9759.2015.10.01. PMID: 27358887; PMCID: PMC4923640.